FIG. 1

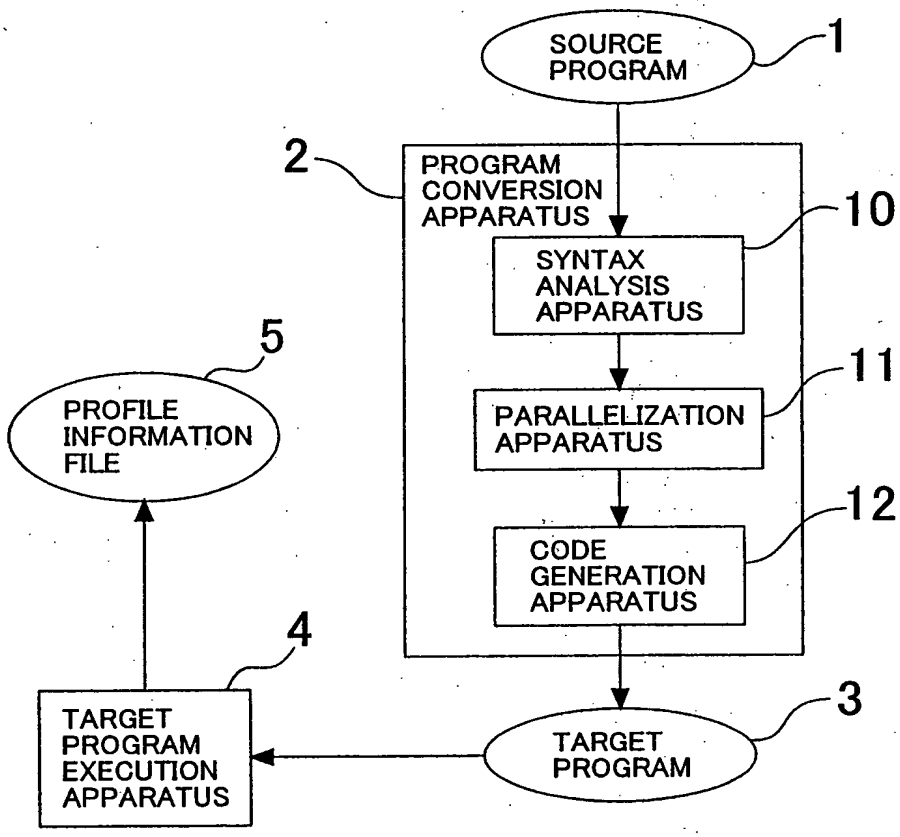


FIG. 2

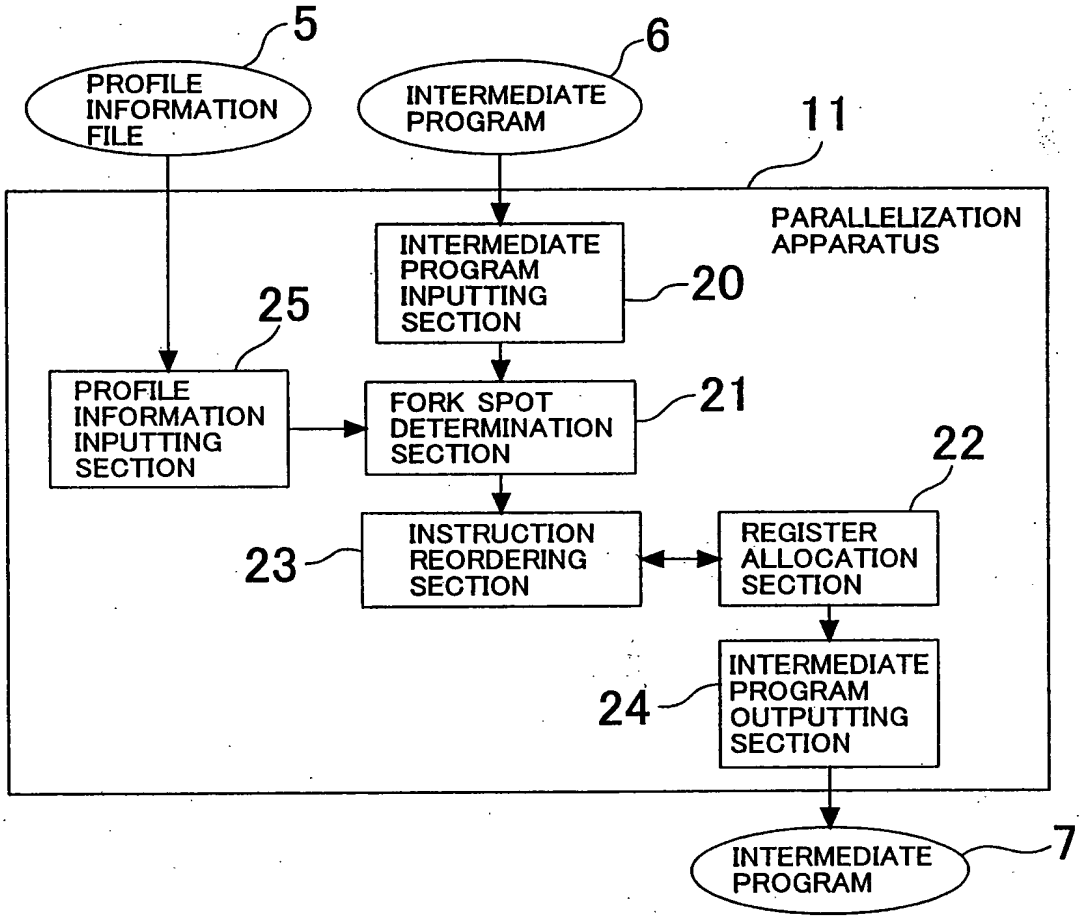

5
PROFILE INFORMATION FILE
6
INTERMEDIATE PROGRAM
11
PARALLELIZATION APPARATUS
INTERMEDIATE PROGRAM INPUTTING SECTION
20
25
PROFILE INFORMATION INPUTTING SECTION
FORK SPOT DETERMINATION SECTION
21
22
23
INSTRUCTION REORDERING SECTION
REGISTER ALLOCATION SECTION
INTERMEDIATE PROGRAM OUTPUTTING SECTION
24
INTERMEDIATE PROGRAM
7

FIG. 3

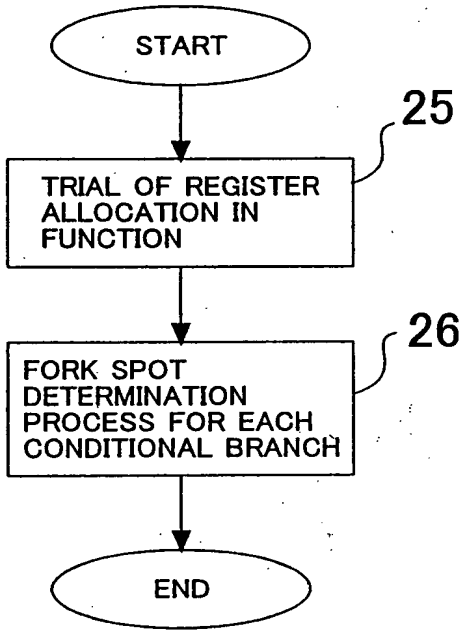

START
25
TRIAL OF REGISTER ALLOCATION IN FUNCTION
26
FORK SPOT DETERMINATION PROCESS FOR EACH CONDITIONAL BRANCH
END

FIG. 5

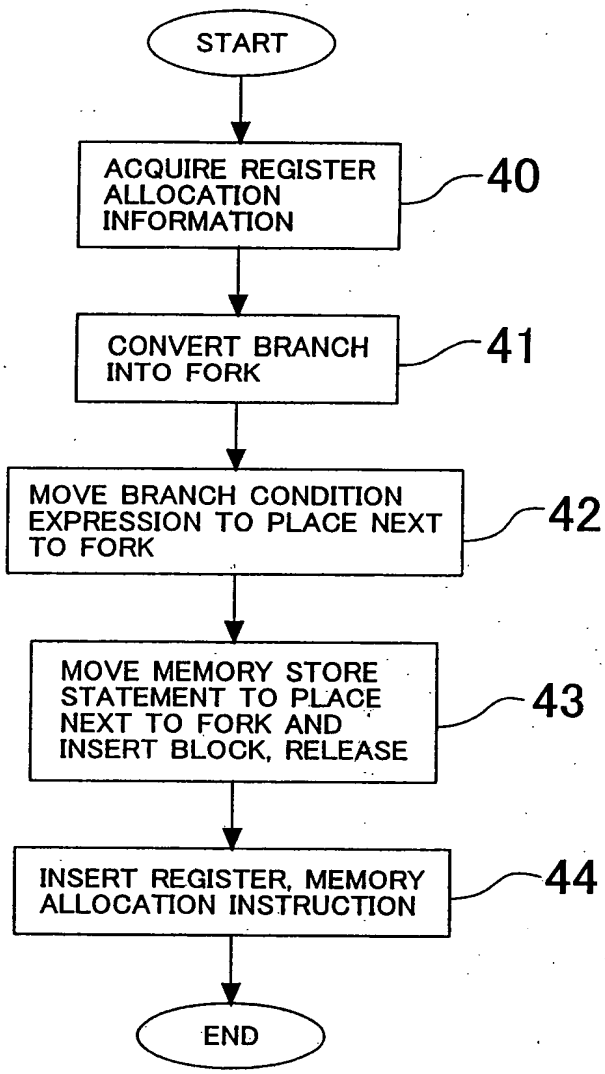

START
ACQUIRE REGISTER ALLOCATION INFORMATION
40
CONVERT BRANCH INTO FORK
41
MOVE BRANCH CONDITION EXPRESSION TO PLACE NEXT TO FORK
42
MOVE MEMORY STORE STATEMENT TO PLACE NEXT TO FORK AND INSERT BLOCK, RELEASE
43
INSERT REGISTER, MEMORY ALLOCATION INSTRUCTION
44
END

FIG. 4

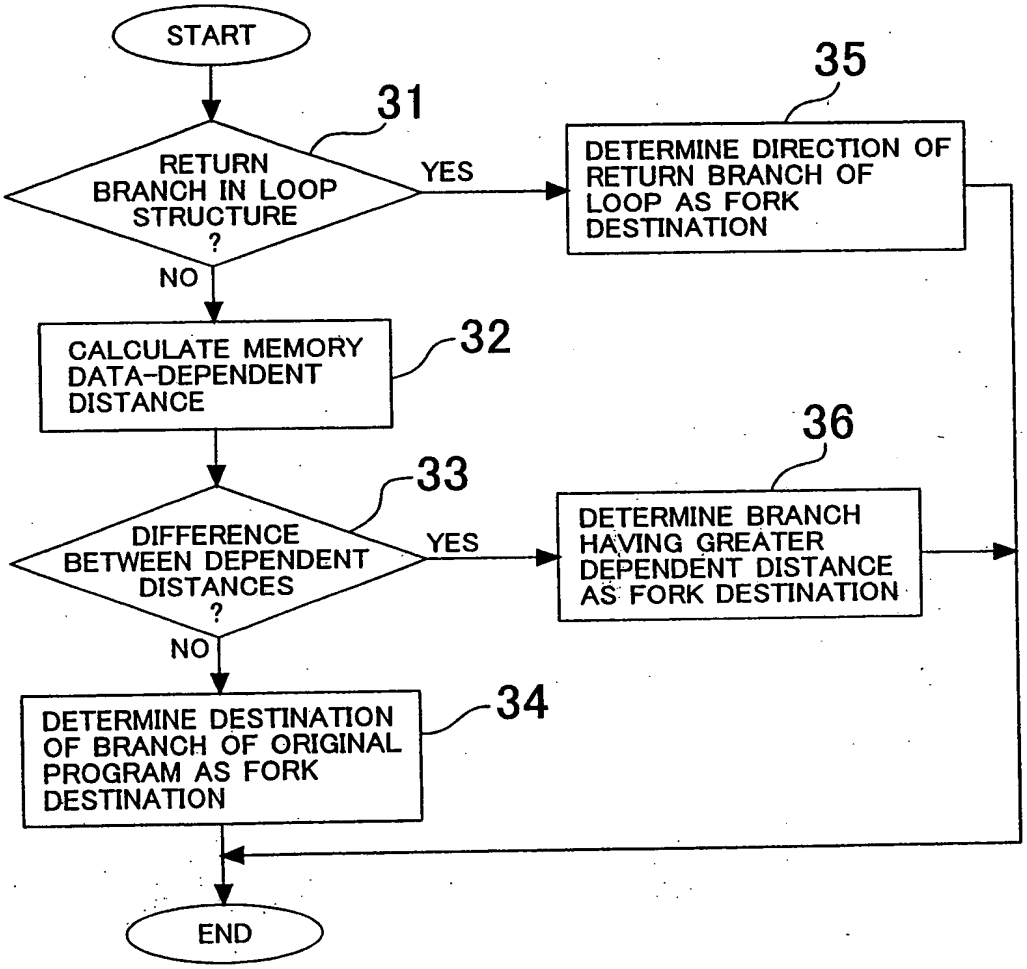

START
31
RETURN BRANCH IN LOOP STRUCTURE ?
YES
35
DETERMINE DIRECTION OF RETURN BRANCH OF LOOP AS FORK DESTINATION
NO
CALCULATE MEMORY DATA-DEPENDENT DISTANCE
32
33
DIFFERENCE BETWEEN DEPENDENT DISTANCES ?
YES
36
DETERMINE BRANCH HAVING GREATER DEPENDENT DISTANCE AS FORK DESTINATION
NO
DETERMINE DESTINATION OF BRANCH OF ORIGINAL PROGRAM AS FORK DESTINATION
34
END

FIG. 6(A)

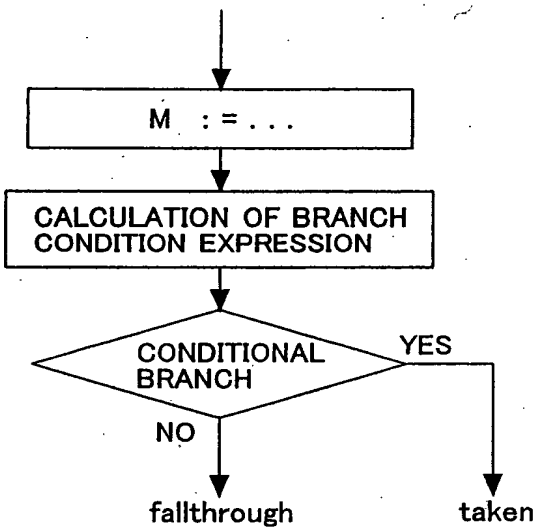

M : = . . .
CALCULATION OF BRANCH CONDITION EXPRESSION
CONDITIONAL BRANCH
YES
NO
fallthrough
taken

FIG. 6(B)

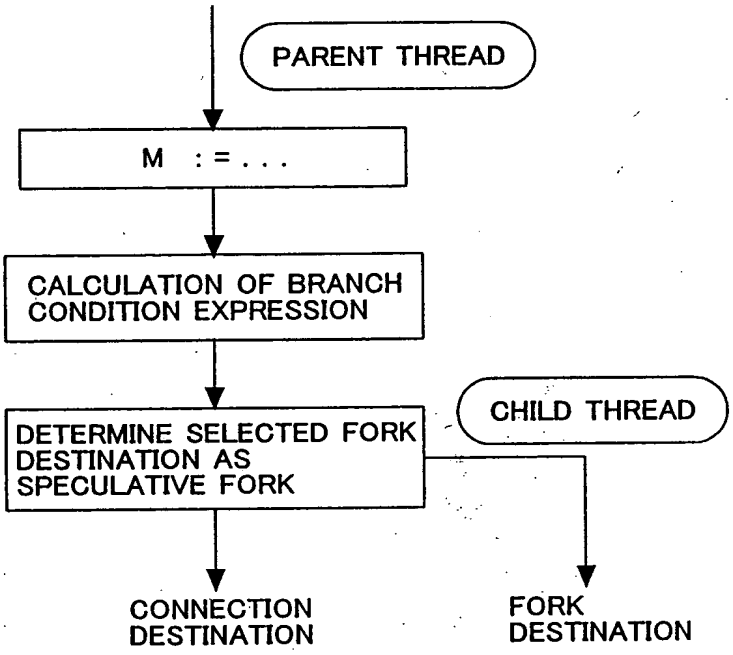

PARENT THREAD
M : = . . .
CALCULATION OF BRANCH CONDITION EXPRESSION
DETERMINE SELECTED FORK DESTINATION AS SPECULATIVE FORK
CHILD THREAD
CONNECTION DESTINATION
FORK DESTINATION

FIG. 6(C)

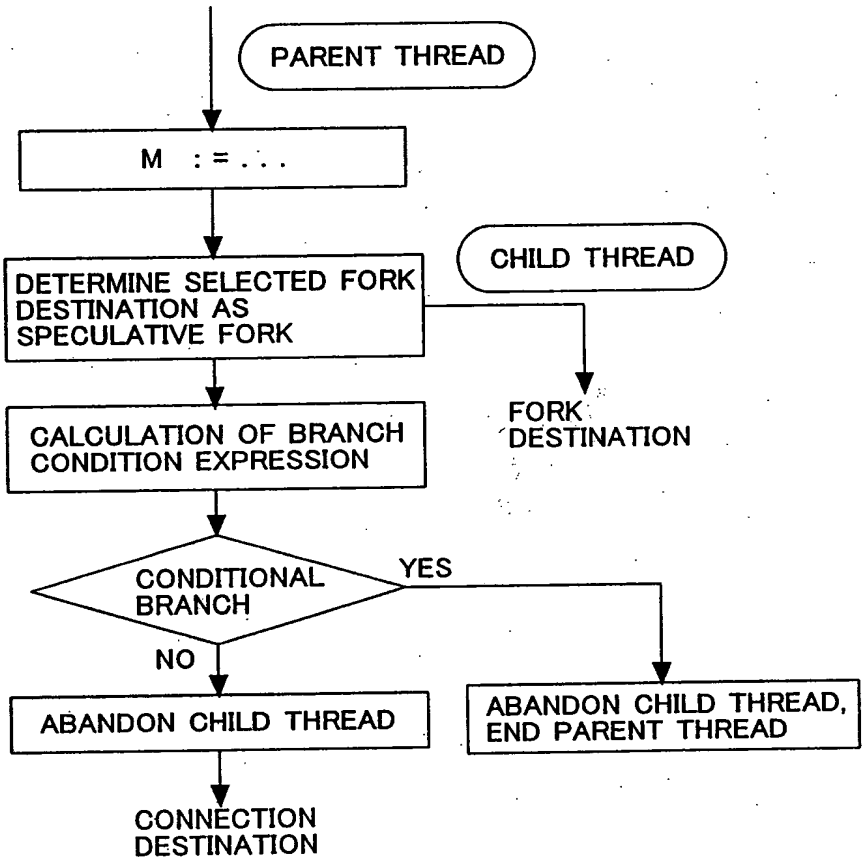

PARENT THREAD
M : = . . .
DETERMINE SELECTED FORK DESTINATION AS SPECULATIVE FORK
CHILD THREAD
FORK DESTINATION
CALCULATION OF BRANCH CONDITION EXPRESSION
CONDITIONAL BRANCH
YES
NO
ABANDON CHILD THREAD
ABANDON CHILD THREAD, END PARENT THREAD
CONNECTION DESTINATION

FIG. 6(D)

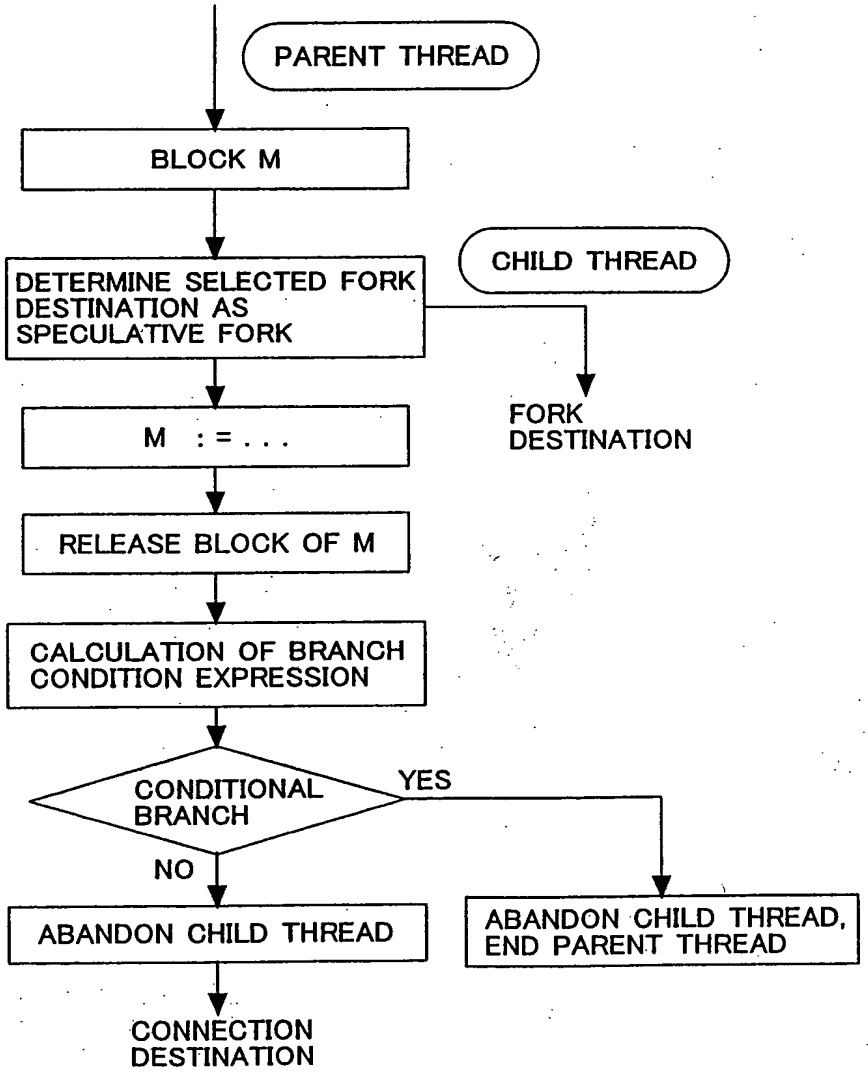

PARENT THREAD
BLOCK M
DETERMINE SELECTED FORK DESTINATION AS SPECULATIVE FORK
CHILD THREAD
FORK DESTINATION
M : = . . .
RELEASE BLOCK OF M
CALCULATION OF BRANCH CONDITION EXPRESSION
CONDITIONAL BRANCH
YES
NO
ABANDON CHILD THREAD
ABANDON CHILD THREAD, END PARENT THREAD
CONNECTION DESTINATION

FIG. 6(E)

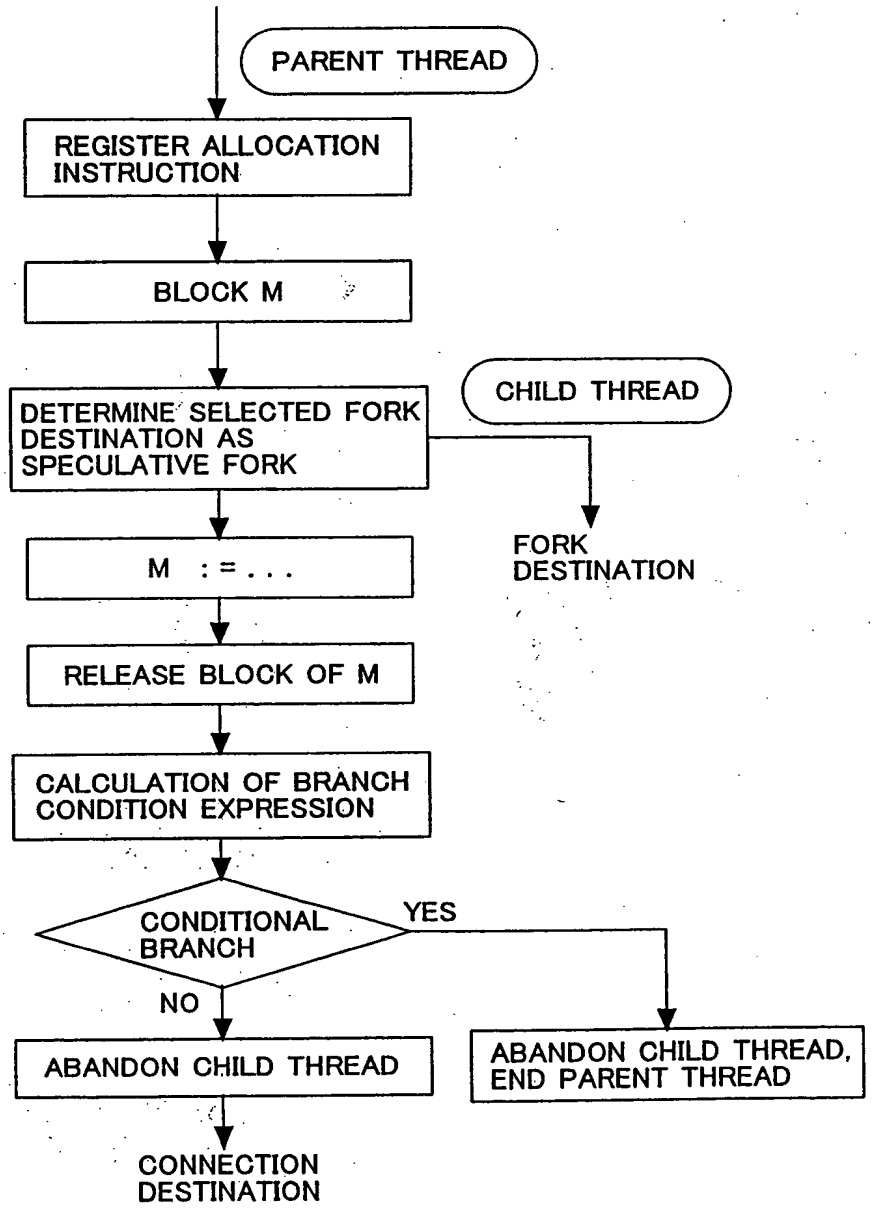

PARENT THREAD
REGISTER ALLOCATION INSTRUCTION
BLOCK M
DETERMINE SELECTED FORK DESTINATION AS SPECULATIVE FORK
CHILD THREAD
FORK DESTINATION
M := . . .
RELEASE BLOCK OF M
CALCULATION OF BRANCH CONDITION EXPRESSION
CONDITIONAL BRANCH
YES
NO
ABANDON CHILD THREAD
ABANDON CHILD THREAD, END PARENT THREAD
CONNECTION DESTINATION

FIG. 7

SPFORK 1	TO CREATE A SPECULATION MODE CHILD THREAD FOR STARTING EXECUTION FROM 1
TTERM c	TO END SELF THREAD AND SETTLE CHILD THREAD IF C IS TRUE
FTERM c	TO END SELF THREAD AND SETTLE CHILD THREAD IF C IS FALSE
THABORT	TO ABANDON A CHILD THREAD OF A SPECULATION MODE
BLACK m	TO DESIGNATE A MEMORY ADDRESS DESIGNATED WITH M AS BLOCK
RELEASE m	TO CLEAR BLOCK SET TO MEMORY ADDRESS DESIGNATED WITH M
DSPIN	TO CREATE A CHILD THREAD CREATED BY SUCCEEDING FORK IN DATA-DEPENDENT SPECULATION MODE
DSPOUT	TO CLEAR DATA-DEPENDENT SPECULATION MODE OF CHILD THREAD
RDCL t, . . .	TO INSTRUCT TO ALLOCATE INTERMEDIATE TERMS/ VARIABLES DESIGNATED WITH t, ··· TO REGISTER
MDCL t, . . .	TO INSTRUCT TO ALLOCATE INTERMEDIATE TERMS/ VARIABLES DESIGNATED WITH t, ··· TO MEMORY

FIG. 8

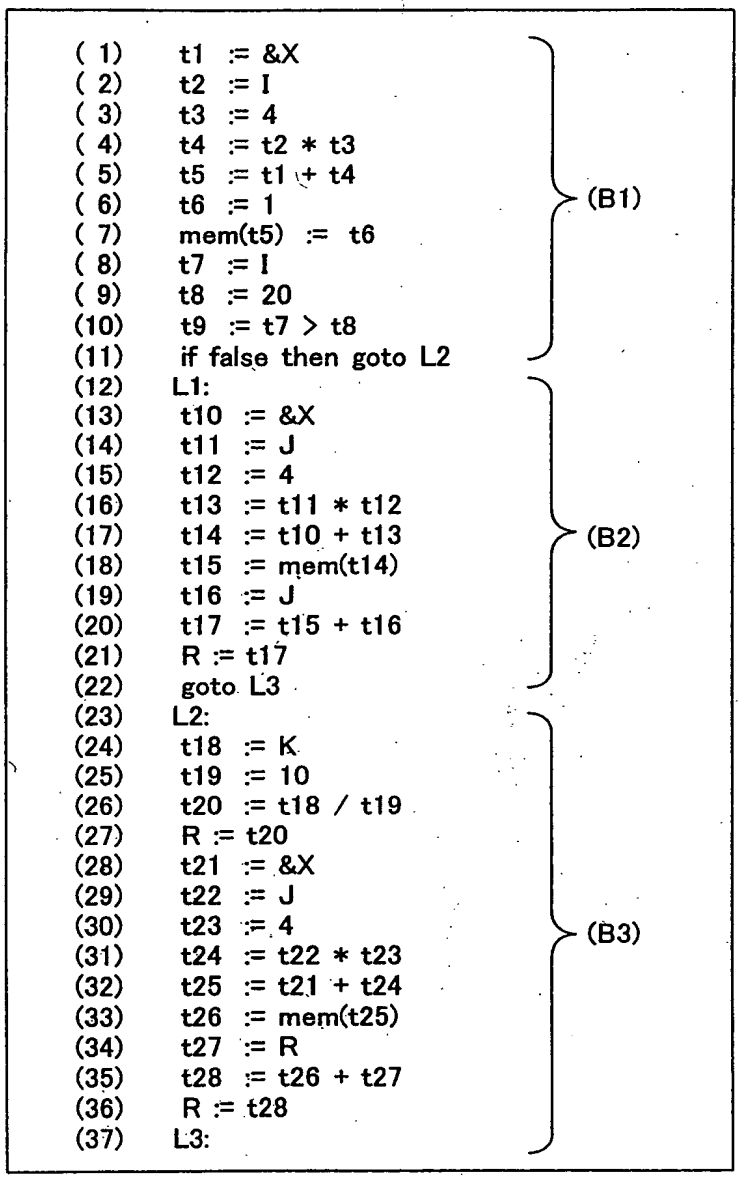

(1) t1 := &X
(2) t2 := I
(3) t3 := 4
(4) t4 := t2 * t3
(5) t5 := t1 + t4
(6) t6 := 1
(7) mem(t5) := t6
(8) t7 := I
(9) t8 := 20
(10) t9 := t7 > t8
(11) if false then goto L2
(B1)
(12) L1:
(13) t10 := &X
(14) t11 := J
(15) t12 := 4
(16) t13 := t11 * t12
(17) t14 := t10 + t13
(18) t15 := mem(t14)
(19) t16 := J
(20) t17 := t15 + t16
(21) R := t17
(22) goto L3
(B2)
(23) L2:
(24) t18 := K
(25) t19 := 10
(26) t20 := t18 / t19
(27) R := t20
(28) t21 := &X
(29) t22 := J
(30) t23 := 4
(31) t24 := t22 * t23
(32) t25 := t21 + t24
(33) t26 := mem(t25)
(34) t27 := R
(35) t28 := t26 + t27
(36) R := t28
(37) L3:
(B3)

FIG. 9

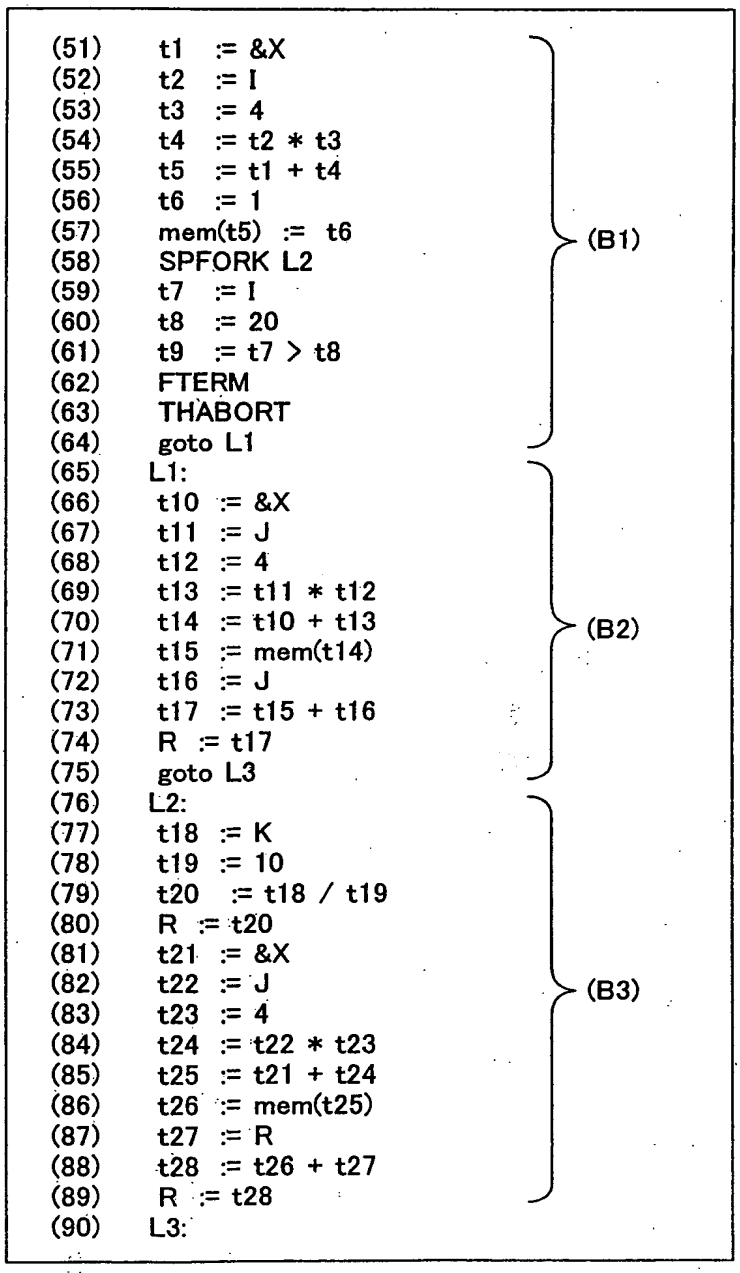

(51) t1 := &X
(52) t2 := I
(53) t3 := 4
(54) t4 := t2 * t3
(55) t5 := t1 + t4
(56) t6 := 1
(57) mem(t5) := t6
(58) SPFORK L2
(59) t7 := I
(60) t8 := 20
(61) t9 := t7 > t8
(62) FTERM
(63) THABORT
(64) goto L1
(65) L1:
(66) t10 := &X
(67) t11 := J
(68) t12 := 4
(69) t13 := t11 * t12
(70) t14 := t10 + t13
(71) t15 := mem(t14)
(72) t16 := J
(73) t17 := t15 + t16
(74) R := t17
(75) goto L3
(76) L2:
(77) t18 := K
(78) t19 := 10
(79) t20 := t18 / t19
(80) R := t20
(81) t21 := &X
(82) t22 := J
(83) t23 := 4
(84) t24 := t22 * t23
(85) t25 := t21 + t24
(86) t26 := mem(t25)
(87) t27 := R
(88) t28 := t26 + t27
(89) R := t28
(90) L3:
(B1)
(B2)
(B3)

FIG. 10

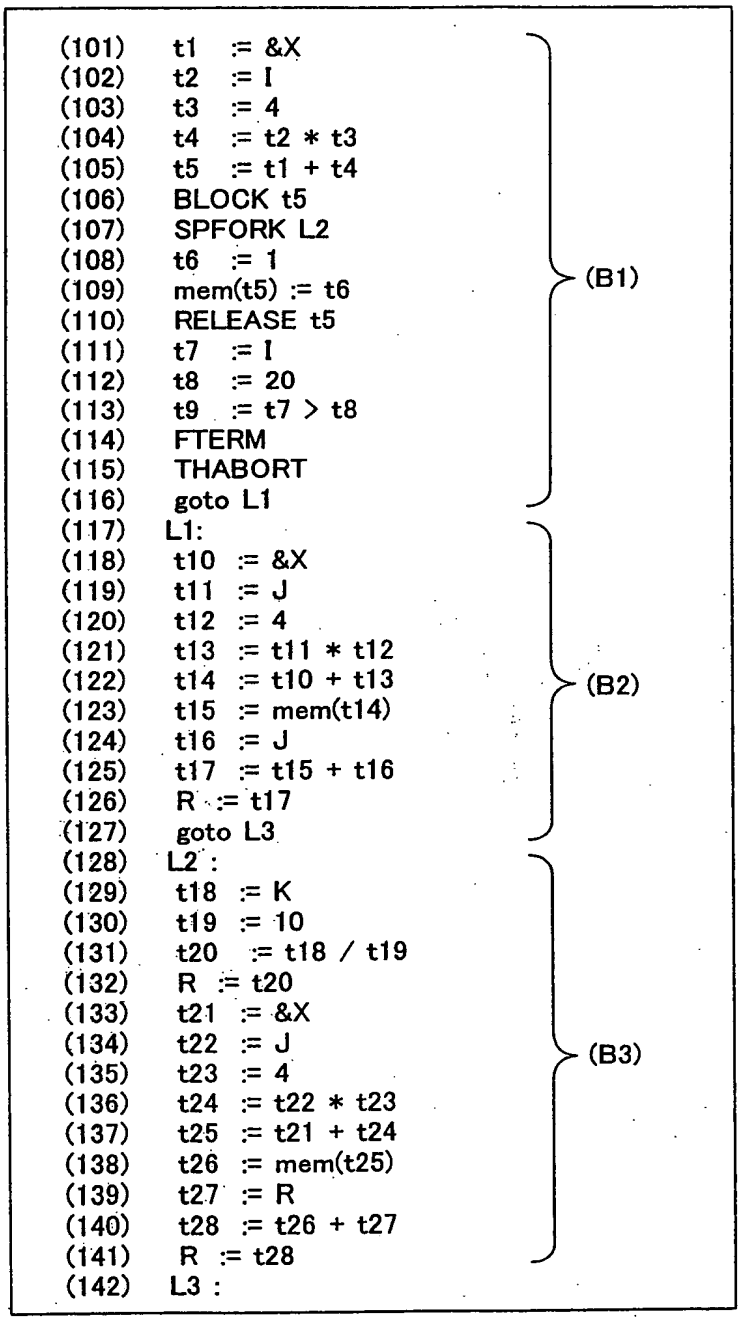

(101) t1 := &X
(102) t2 := I
(103) t3 := 4
(104) t4 := t2 * t3
(105) t5 := t1 + t4
(106) BLOCK t5
(107) SPFORK L2
(108) t6 := 1
(109) mem(t5) := t6
(110) RELEASE t5
(111) t7 := I
(112) t8 := 20
(113) t9 := t7 > t8
(114) FTERM
(115) THABORT
(116) goto L1
(B1)
(117) L1:
(118) t10 := &X
(119) t11 := J
(120) t12 := 4
(121) t13 := t11 * t12
(122) t14 := t10 + t13
(123) t15 := mem(t14)
(124) t16 := J
(125) t17 := t15 + t16
(126) R := t17
(127) goto L3
(B2)
(128) L2 :
(129) t18 := K
(130) t19 := 10
(131) t20 := t18 / t19
(132) R := t20
(133) t21 := &X
(134) t22 := J
(135) t23 := 4
(136) t24 := t22 * t23
(137) t25 := t21 + t24
(138) t26 := mem(t25)
(139) t27 := R
(140) t28 := t26 + t27
(141) R := t28
(B3)
(142) L3 :

FIG. 11

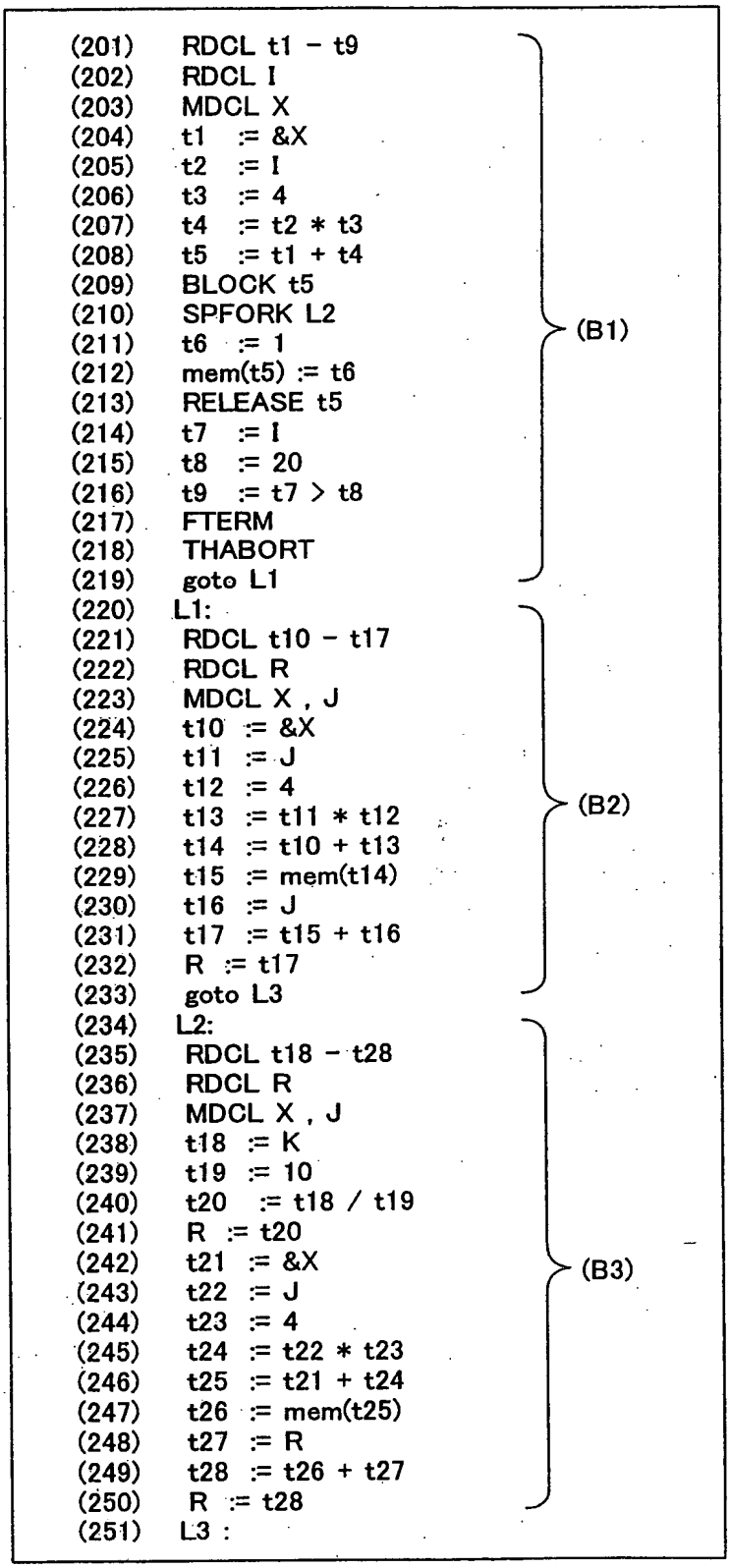

(201) RDCL t1 - t9
(202) RDCL I
(203) MDCL X
(204) t1 := &X
(205) t2 := I
(206) t3 := 4
(207) t4 := t2 * t3
(208) t5 := t1 + t4
(209) BLOCK t5
(210) SPFORK L2
(211) t6 := 1
(212) mem(t5) := t6
(213) RELEASE t5
(214) t7 := I
(215) t8 := 20
(216) t9 := t7 > t8
(217) FTERM
(218) THABORT
(219) goto L1
(220) L1:
(221) RDCL t10 - t17
(222) RDCL R
(223) MDCL X , J
(224) t10 := &X
(225) t11 := J
(226) t12 := 4
(227) t13 := t11 * t12
(228) t14 := t10 + t13
(229) t15 := mem(t14)
(230) t16 := J
(231) t17 := t15 + t16
(232) R := t17
(233) goto L3
(234) L2:
(235) RDCL t18 - t28
(236) RDCL R
(237) MDCL X , J
(238) t18 := K
(239) t19 := 10
(240) t20 := t18 / t19
(241) R := t20
(242) t21 := &X
(243) t22 := J
(244) t23 := 4
(245) t24 := t22 * t23
(246) t25 := t21 + t24
(247) t26 := mem(t25)
(248) t27 := R
(249) t28 := t26 + t27
(250) R := t28
(251) L3 :
(B1)
(B2)
(B3)

FIG. 12

```
(255)    r21  := &X
(256)    r22  := r11
(257)    r23  := 4
(258)    r24  := r22 * r23
(259)    r25  := r21 + r24
(260)    BLOCK r25
(261)    SPFORK L2
(262)    r26  := 1
(263)    mem(r25) := r26
(264)    RELEASE r25
(265)    r27  := r11
(266)    r28  := 20
(267)    r29  := r27 > r28
(268)    FTERM r29
(269)    THABORT
(270)    goto L1
(271)   L1:
(272)    r20  := &X
(273)    r21  := mem(&J)
(274)    r22  := 4
(275)    r23  := r21 * r22
(276)    r24  := r20 + r23
(277)    r25  := mem(r24)
(278)    r26  := mem(&J)
(279)    r27  := r25 + r26
(280)    r12  := r27
(281)    goto L3
(282)   L2:
(283)    r20  := r13
(284)    r21  := 10
(285)    r22  := r20 / r21
(286)    r12  := r22
(287)    r23  := &X
(288)    r24  := mem(&J)
(289)    r25  := 4
(290)    r26  := r24 * r25
(291)    r27  := r23 + r26
(292)    r28  := mem(r27)
(293)    r29  := r12
(294)    r30  := r28 + r29
(295)    r12  := r30
(296)   L3:
```

FIG. 13

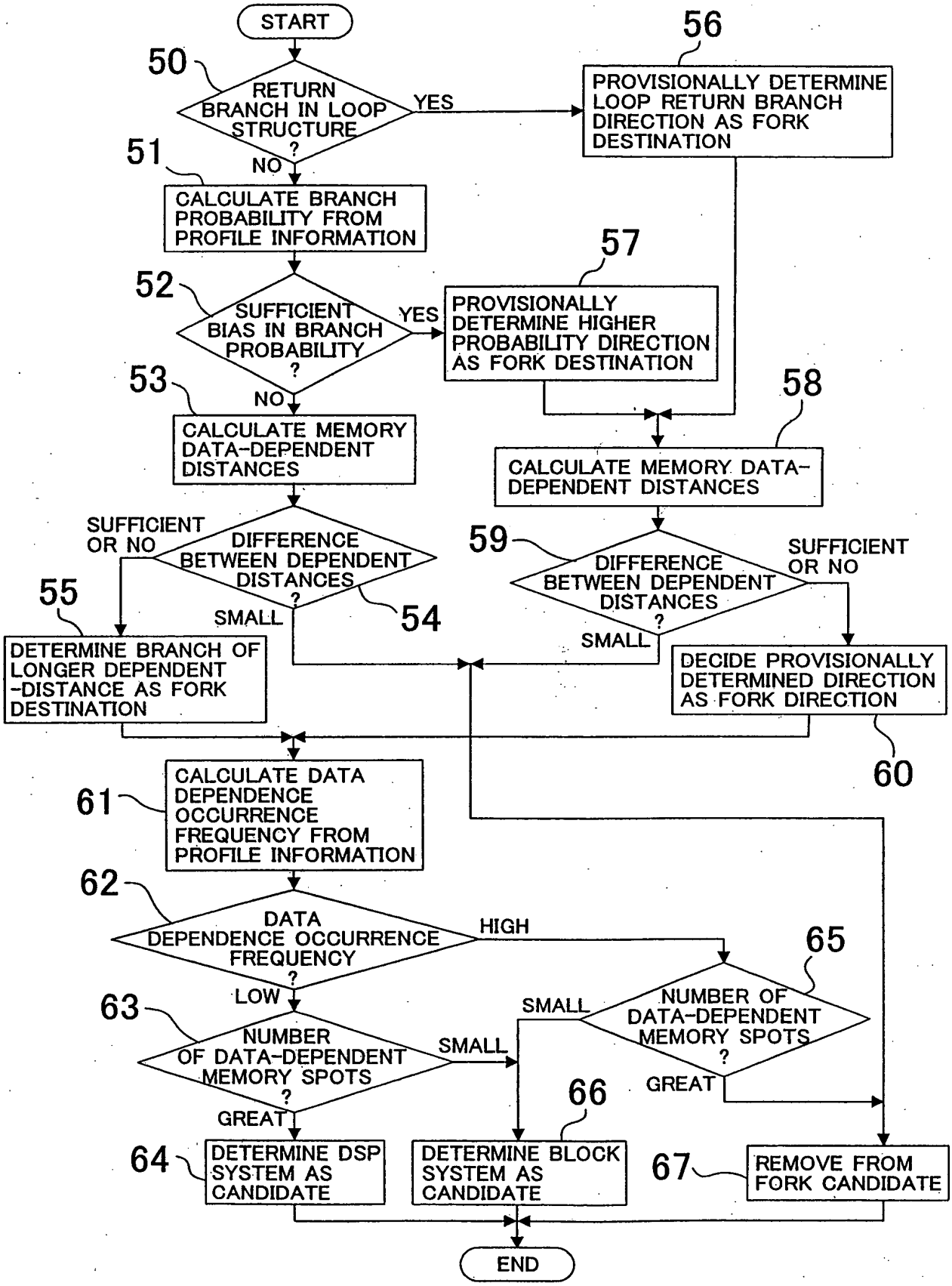

START
50
RETURN BRANCH IN LOOP STRUCTURE ?
YES
56
PROVISIONALLY DETERMINE LOOP RETURN BRANCH DIRECTION AS FORK DESTINATION
NO
51
CALCULATE BRANCH PROBABILITY FROM PROFILE INFORMATION
52
SUFFICIENT BIAS IN BRANCH PROBABILITY ?
YES
57
PROVISIONALLY DETERMINE HIGHER PROBABILITY DIRECTION AS FORK DESTINATION
NO
53
CALCULATE MEMORY DATA-DEPENDENT DISTANCES
58
CALCULATE MEMORY DATA-DEPENDENT DISTANCES
SUFFICIENT OR NO
DIFFERENCE BETWEEN DEPENDENT DISTANCES ?
54
SMALL
59
DIFFERENCE BETWEEN DEPENDENT DISTANCES ?
SUFFICIENT OR NO
SMALL
55
DETERMINE BRANCH OF LONGER DEPENDENT -DISTANCE AS FORK DESTINATION
DECIDE PROVISIONALLY DETERMINED DIRECTION AS FORK DIRECTION
60
61
CALCULATE DATA DEPENDENCE OCCURRENCE FREQUENCY FROM PROFILE INFORMATION
62
DATA DEPENDENCE OCCURRENCE FREQUENCY ?
HIGH
65
NUMBER OF DATA-DEPENDENT MEMORY SPOTS ?
SMALL
GREAT
LOW
63
NUMBER OF DATA-DEPENDENT MEMORY SPOTS ?
SMALL
66
GREAT
64
DETERMINE DSP SYSTEM AS CANDIDATE
DETERMINE BLOCK SYSTEM AS CANDIDATE
67
REMOVE FROM FORK CANDIDATE
END

FIG. 14

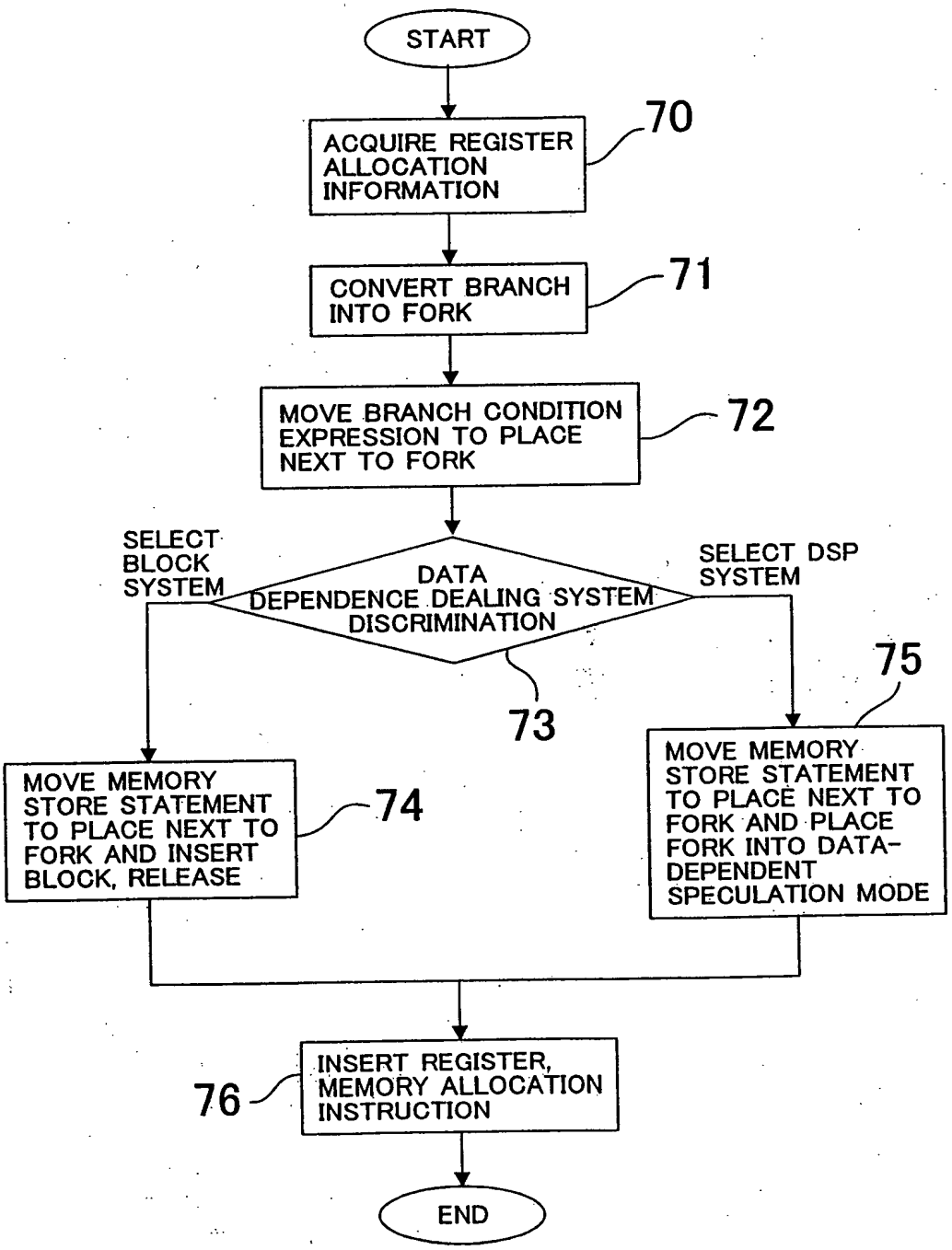

START
ACQUIRE REGISTER ALLOCATION INFORMATION
70
CONVERT BRANCH INTO FORK
71
MOVE BRANCH CONDITION EXPRESSION TO PLACE NEXT TO FORK
72
SELECT BLOCK SYSTEM
DATA DEPENDENCE DEALING SYSTEM DISCRIMINATION
SELECT DSP SYSTEM
73
MOVE MEMORY STORE STATEMENT TO PLACE NEXT TO FORK AND INSERT BLOCK, RELEASE
74
75
MOVE MEMORY STORE STATEMENT TO PLACE NEXT TO FORK AND PLACE FORK INTO DATA-DEPENDENT SPECULATION MODE
INSERT REGISTER, MEMORY ALLOCATION INSTRUCTION
76
END

FIG. 15

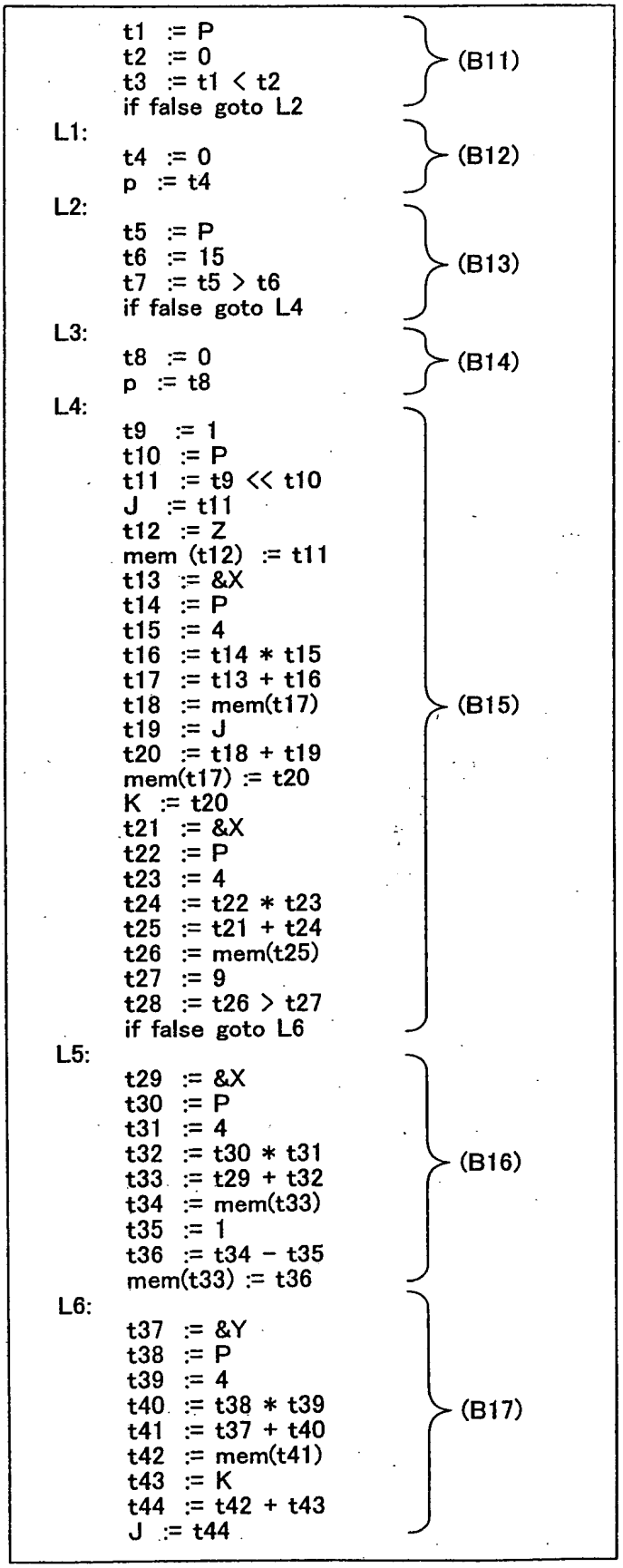

t1 := P
t2 := 0
t3 := t1 < t2
if false goto L2
(B11)
L1:
t4 := 0
p := t4
(B12)
L2:
t5 := P
t6 := 15
t7 := t5 > t6
if false goto L4
(B13)
L3:
t8 := 0
p := t8
(B14)
L4:
t9 := 1
t10 := P
t11 := t9 << t10
J := t11
t12 := Z
mem (t12) := t11
t13 := &X
t14 := P
t15 := 4
t16 := t14 * t15
t17 := t13 + t16
t18 := mem(t17)
t19 := J
t20 := t18 + t19
mem(t17) := t20
K := t20
t21 := &X
t22 := P
t23 := 4
t24 := t22 * t23
t25 := t21 + t24
t26 := mem(t25)
t27 := 9
t28 := t26 > t27
if false goto L6
(B15)
L5:
t29 := &X
t30 := P
t31 := 4
t32 := t30 * t31
t33 := t29 + t32
t34 := mem(t33)
t35 := 1
t36 := t34 - t35
mem(t33) := t36
(B16)
L6:
t37 := &Y
t38 := P
t39 := 4
t40 := t38 * t39
t41 := t37 + t40
t42 := mem(t41)
t43 := K
t44 := t42 + t43
J := t44
(B17)

FIG. 16(A)

BRANCHING NUMBER

B I1	B I2: 2D	B I3: I8D
B I3	B I4: 3D	B I5: I7D
B I5	B I6: 3D	B I7: I7D

FIG. 16(B)

MEMORY DATA DEPENDENCE

B I5 → B I6	I2D
B I5 → B I7	4

FIG. 17

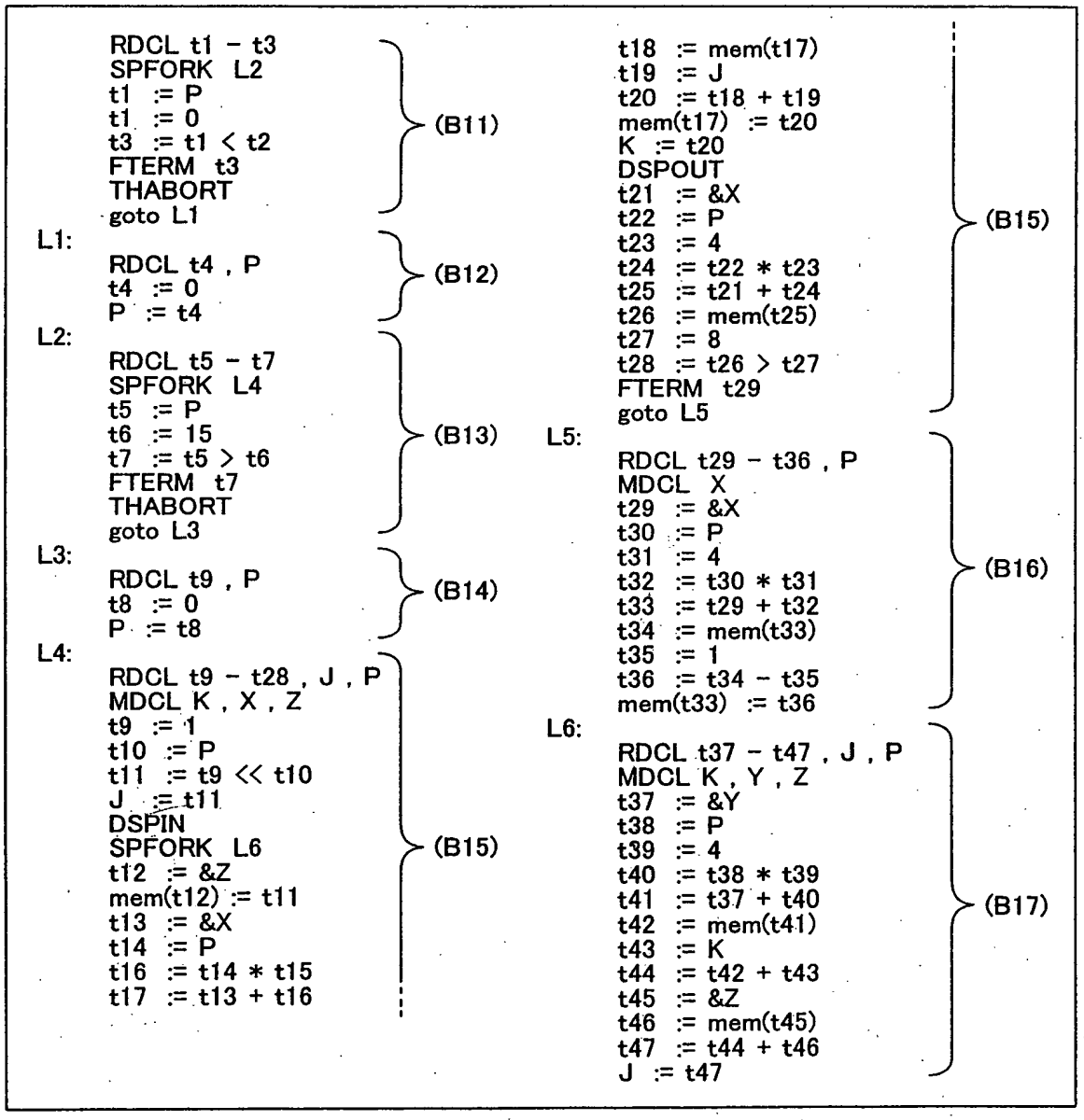

RDCL t1 - t3
SPFORK L2
t1 := P
t1 := 0
t3 := t1 < t2
FTERM t3
THABORT
goto L1
(B11)
L1:
RDCL t4 , P
t4 := 0
P := t4
(B12)
L2:
RDCL t5 - t7
SPFORK L4
t5 := P
t6 := 15
t7 := t5 > t6
FTERM t7
THABORT
goto L3
(B13)
L3:
RDCL t9 , P
t8 := 0
P := t8
(B14)
L4:
RDCL t9 - t28 , J , P
MDCL K , X , Z
t9 := 1
t10 := P
t11 := t9 << t10
J := t11
DSPIN
SPFORK L6
t12 := &Z
mem(t12) := t11
t13 := &X
t14 := P
t16 := t14 * t15
t17 := t13 + t16
(B15)
t18 := mem(t17)
t19 := J
t20 := t18 + t19
mem(t17) := t20
K := t20
DSPOUT
t21 := &X
t22 := P
t23 := 4
t24 := t22 * t23
t25 := t21 + t24
t26 := mem(t25)
t27 := 8
t28 := t26 > t27
FTERM t29
goto L5
(B15)
L5:
RDCL t29 - t36 , P
MDCL X
t29 := &X
t30 := P
t31 := 4
t32 := t30 * t31
t33 := t29 + t32
t34 := mem(t33)
t35 := 1
t36 := t34 - t35
mem(t33) := t36
(B16)
L6:
RDCL t37 - t47 , J , P
MDCL K , Y , Z
t37 := &Y
t38 := P
t39 := 4
t40 := t38 * t39
t41 := t37 + t40
t42 := mem(t41)
t43 := K
t44 := t42 + t43
t45 := &Z
t46 := mem(t45)
t47 := t44 + t46
J := t47
(B17)

FIG. 18

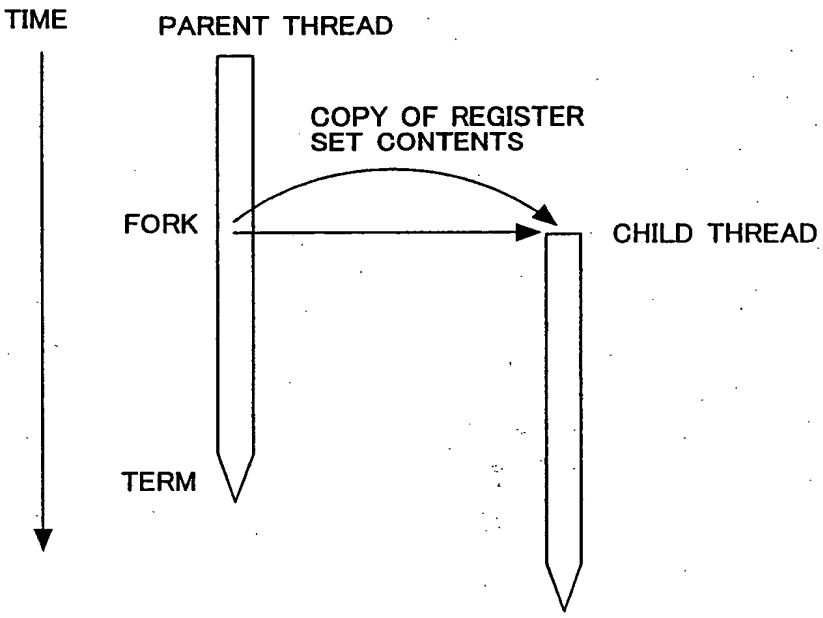

TIME
PARENT THREAD
COPY OF REGISTER SET CONTENTS
FORK
CHILD THREAD
TERM

FIG. 19

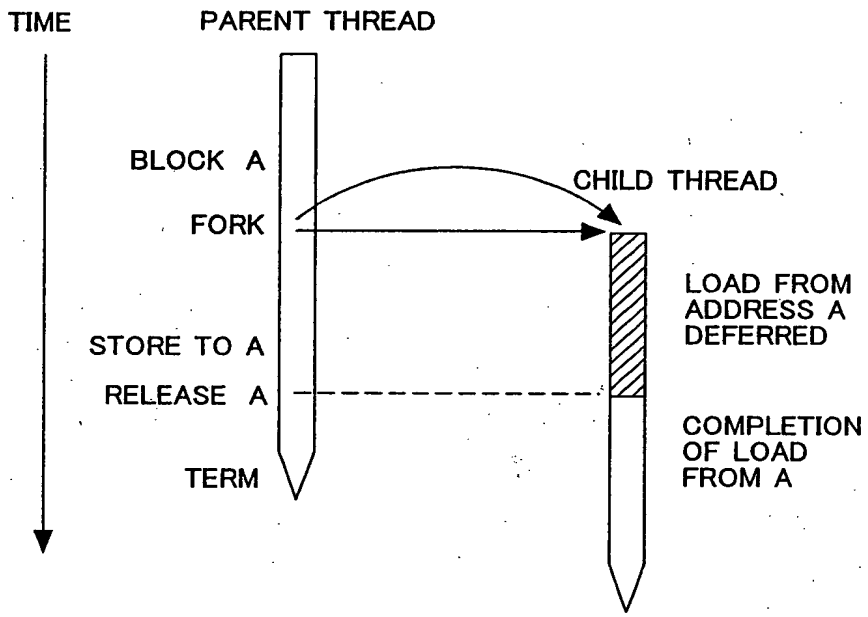

TIME
PARENT THREAD
BLOCK A
FORK
CHILD THREAD
LOAD FROM ADDRESS A DEFERRED
STORE TO A
RELEASE A
COMPLETION OF LOAD FROM A
TERM

FIG. 20(A)

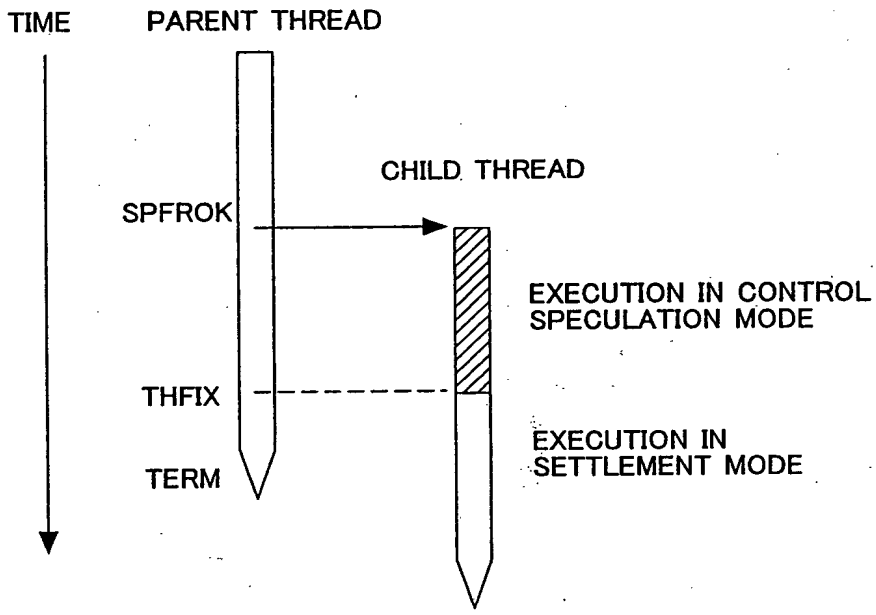

TIME
PARENT THREAD
CHILD THREAD
SPFROK
EXECUTION IN CONTROL SPECULATION MODE
THFIX
EXECUTION IN SETTLEMENT MODE
TERM

FIG. 20(B)

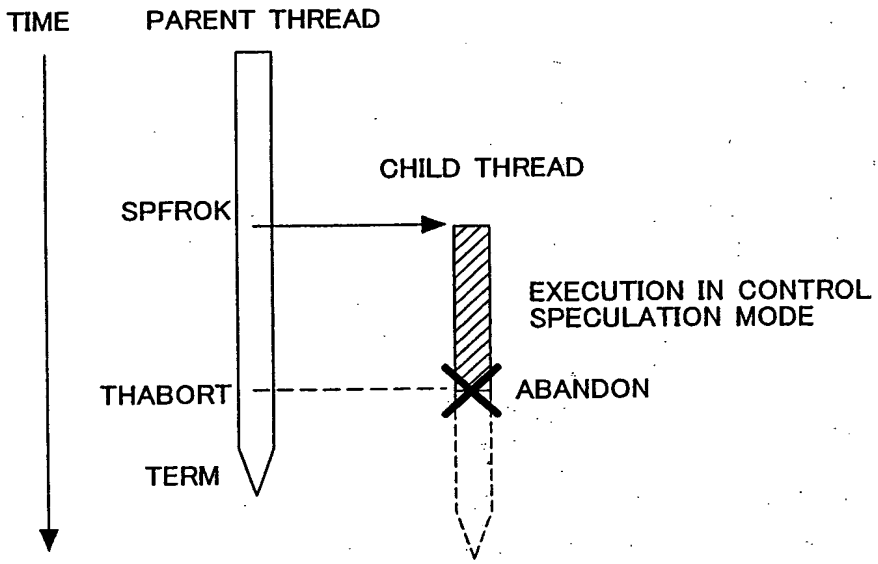
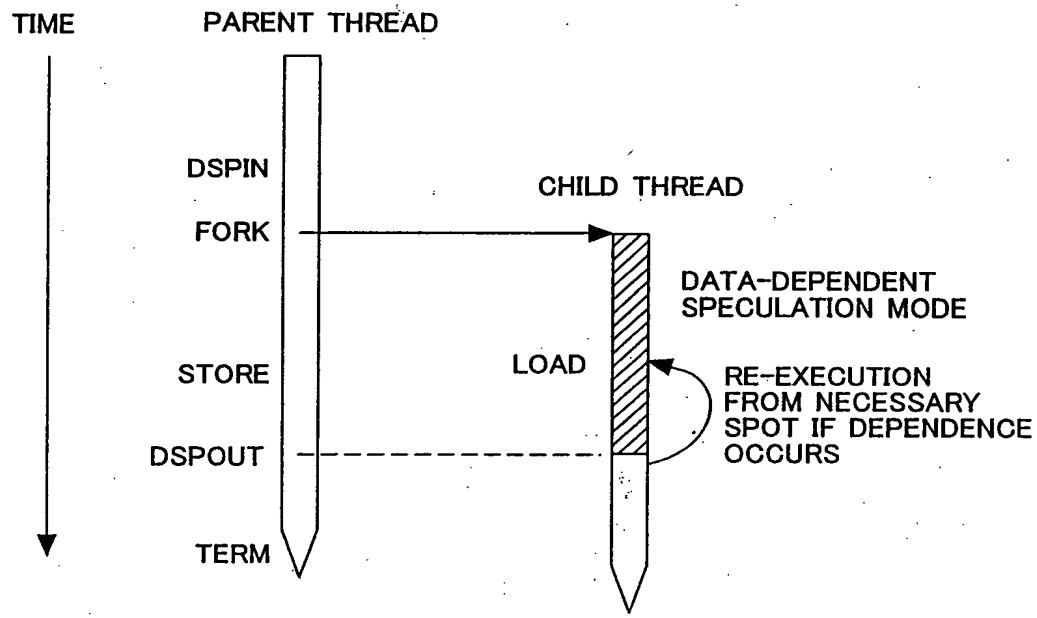

TIME
PARENT THREAD
CHILD THREAD
SPFROK
EXECUTION IN CONTROL SPECULATION MODE
THABORT
ABANDON
TERM

FIG. 21


TIME
PARENT THREAD
DSPIN
FORK
CHILD THREAD
DATA-DEPENDENT
SPECULATION MODE
STORE
LOAD
RE-EXECUTION
FROM NECESSARY
SPOT IF DEPENDENCE
OCCURS
DSPOUT
TERM